

Сравнительный анализ аппаратных архитектур декодера LDPC кодов для систем радиосвязи IEEE 802.11ad

А.А. Шевченко^{1,2}, Р.О. Масленников², А.А. Мальцев¹, М.В. Пантелеев^{1,2},

А.Ф. Ключев^{1,2}, А.Г. Ветчинкин^{1,2}

¹Нижегородский государственный университет им. Н.И.Лобачевского, ²ООО «Радио Гигабит», andrey.shevchenko@wcc.unn.ru

Аннотация — В работе рассматриваются различные подходы к разработке декодера LDPC кода стандарта радиосвязи IEEE 802.11ad беспроводных локальных сетей, производится анализ эффективности трех базовых архитектур, позволяющих достигать различных соотношений между пропускной способностью и величиной требуемых аппаратных ресурсов при реализации декодера на ПЛИС. Показано, что применение полностью параллельной архитектуры позволяет достигать пропускной способности 6.5 Гбит/с и является наиболее аппаратно-эффективным решением, то есть обеспечивает наилучшее отношение величины пропускной способности к объему используемых аппаратных ресурсов. Использование слоистой архитектуры декодера позволяет уменьшить число требуемых аппаратных ресурсов на 46% относительно полностью параллельной архитектуры, но при этом аппаратная эффективность данной архитектуры ниже на 42% и достигается пропускная способность равная 2.1 Гбит/с. Свойства квазицикличности проверочной матрицы рассматриваемого кода позволяет использовать последовательно-параллельную архитектуру, которая требует для реализации только 11% от числа аппаратных ресурсов для полностью параллельной архитектуры, но достигая при этом существенно меньшей пропускной способности в 159 Мбит/с.

Ключевые слова — LDPC коды, аппаратная архитектура декодера, IEEE 802.11ad.

I. ВВЕДЕНИЕ

Коды с малой плотностью проверок на четность (Low Density Parity Check - LDPC) нашли широкое применение в современных системах передачи информации благодаря хорошей помехоустойчивости, относительно невысокой вычислительной сложности и высокой степени параллелизма алгоритмов декодирования, делающих их привлекательными для различных высокоскоростных систем связи. Относительным недостатком LDPC кодов, в общем случае, является ограниченное число доступных вариантов архитектур декодера, что не позволяет разработчикам иметь достаточную гибкость при достижении компромисса между требуемой пропускной способностью и объемом используемых аппаратных ресурсов (числом логических элементов на ПЛИС или занимаемой площадью на кристалле СБИС). Известно, что для всех классов LDPC кодов доступны две предельные архитектуры декодера: полностью параллельная, обеспечивающая

наибольшую пропускную способность, но требующая наибольшего числа аппаратных ресурсов, и полностью последовательная, использующая наименьшее количество ресурсов и обладающая наименьшей производительностью, недостаточной для большинства практических применений.

Однако такое ограничение в полной мере справедливо только для случайных и квази-случайных LDPC кодов. Специально разрабатываемые структурированные LDPC коды лишены данного недостатка и позволяют получать дополнительные промежуточные решения для аппаратных архитектур декодера. Данный класс кодов также носит название аппаратно-ориентированных. Наиболее привлекательным подклассом данной группы являются LDPC коды с непересекающимися слоями, позволяющие использовать модифицированный алгоритм декодирования, характеризующийся лучшей сходимостью [3], а также LDPC коды, обладающие свойством квазицикличности проверочной матрицы [4].

В данной работе рассмотрен LDPC код стандарта IEEE 802.11ad для беспроводных локальных сетей миллиметрового диапазона 60 ГГц [1], обеспечивающий передачу данных между мобильными устройствами со скоростью до 7 Гбит/с. Структура и свойства используемого кода, включающие одновременно наличие нескольких непересекающихся слоев и квазицикличность проверочной матрицы, позволяют одновременно достигать высокой пропускной способности и эффективной аппаратной реализации. В работе исследованы основные доступные варианты архитектур декодера LDPC кода со скоростью кода $\frac{1}{2}$ из семейства кодов стандарта IEEE 802.11ad, произведена оценка пропускной способности и представлены результаты оценки требуемых аппаратных ресурсов при их реализации на программируемых логических интегральных схемах (ПЛИС).

II. LDPC КОД СТАНДАРТА IEEE 802.11AD И ИСПОЛЬЗУЕМЫЕ АЛГОРИТМЫ ДЕКОДИРОВАНИЯ

A. Введение в LDPC коды

LDPC коды являются линейными блоковыми кодами и задаются проверочной матрицей \mathbf{H} размерностью $M \times N$, обладающей свойством разреженности, таким, что число ненулевых элементов этой матрицы много меньше общего числа ее элементов. Любой ли-

нейный блочный код может быть представлен в виде графа Таннера, являющегося графической интерпретацией проверочной матрицы \mathbf{H} размерностью $M \times N$. На рис. 1 приведен пример граф Таннера для кода с $M = 4$ и $N = 8$ или $(8, 4)$ кода.

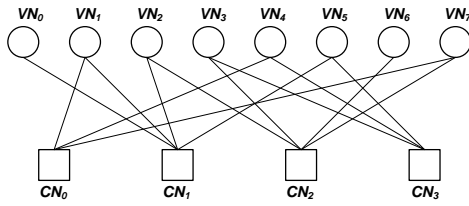


Рис. 1. Граф Таннера проверочной матрицы $(8,4)$ кода

Граф Таннера является двудольным графом, вершины которого делятся на два множества: 1) N кодовых вершин (узлов), соответствующих столбцам проверочной матрицы \mathbf{H} ; 2) M проверочных вершин (узлов), соответствующих строкам проверочной матрицы \mathbf{H} . На Рис. 1 кодовые узлы изображены круглыми маркерами (VN_i), проверочные узлы изображены квадратными маркерами (CN_i). Ребра, соединяющие вершины графа, соответствуют ненулевым позициям в матрице \mathbf{H} .

В. Квазициклический LDPC код стандарта IEEE 802.11ad

Проверочная матрица квазициклического LDPC кода (QC-LDPC) задается с помощью меньшей по размеру матрицы $M' \times N'$ (так называемой макроматрицы). Каждый элемент макроматрицы соответствует элементарной двоичной матрице размером $Z \times Z$, которая может быть либо единичной циклически сдвинутой вправо на некоторую величину от 0 до $Z - 1$, либо нулевой матрицей аналогичного размера. Таким образом, результирующая проверочная матрица \mathbf{H} кода имеет размер $M'Z \times N'Z$ и является конкатенацией элементарных матриц размером $Z \times Z$. Длина кодового слова в этом случае равняется $N'Z$ бит, а скорость кода соответственно равна $(N' - M') / N'$.

Рассматриваемый QC-LDPC код стандарта IEEE 802.11ad задается макроматрицей 8×16 с размером элементарной матрицы 42×42 , представленной на Рис. 2. Нулевые элементарные матрицы обозначены как "-1", а ненулевые – величиной соответствующего циклического сдвига вправо единичной матрицы. Результирующая проверочная матрица имеет размер 336×672 и определяет код с длиной кодового слова 672 бита и скоростью кода $1/2$.

Матрица \mathbf{H} рассматриваемого кода характеризуется свойством слоистости, т.е. все строки матрицы могут быть разделены на четыре подматрицы (слоя) (см. рис. 2) таким образом, что столбец каждой подматрицы имеет вес не более чем 1. Данное свойство позволяет использовать модифицированный LBP алгоритм декодирования (см. раздел 2.D), что дает два преимущества при аппаратной реализации декодера. Во-первых, улучшается скорость сходимости, требуя меньшее число итераций по сравнению с традиционным BP алгоритмом декодирования (также см. раздел 2.C). Во-вторых, использование данного алгоритма открывает возможность применения слоистой архитектуры деко-

дера, которая позволяет уменьшить требуемый объем аппаратных ресурсов за счет переиспользования их между подитерациями алгоритма (см. раздел 3.B).

Layer 1	40	-1	38	-1	13	-1	5	-1	18	-1	-1	-1	-1	-1	-1	-1	Layer 2
	34	-1	35	-1	27	-1	-1	30	2	1	-1	-1	-1	-1	-1	-1	
	-1	36	-1	31	-1	7	-1	34	-1	10	41	-1	-1	-1	-1	-1	Layer 4
	-1	27	-1	18	-1	12	20	-1	-1	-1	15	6	-1	-1	-1	-1	
Layer 3	35	-1	41	-1	40	-1	39	-1	28	-1	-1	3	28	-1	-1	-1	
	29	-1	0	-1	-1	22	-1	4	-1	28	-1	27	-1	23	-1	-1	
	-1	31	-1	23	-1	21	-1	20	-1	-1	12	-1	-1	0	13	-1	
	-1	22	-1	34	31	-1	14	-1	4	-1	-1	-1	13	-1	22	24	

Рис. 2. Проверочная матрица IEEE 802.11ad QC-LDPC кода со скоростью кода $1/2$

Анализ, выполненный в данной работе, полностью сфокусирован на возможностях аппаратной реализации декодера для рассматриваемого кода. Свойства данного кода с точки зрения помехоустойчивости хорошо изучены, и соответствующие характеристики могут быть найдены в источниках [5] и [7].

С. Алгоритм декодирования Belief Propagation (BP)

Наиболее распространенным алгоритмом декодирования LDPC кодов является итеративный алгоритм, основанный на пересчете апостериорных вероятностей бит кодового слова (так называемый Belief Propagation алгоритм – BP-алгоритм). Алгоритм минимизирует вероятность ошибки для каждого бита c_n кодового слова \mathbf{c} , учитывая аддитивный вклад наблюдения как n -го бита, так и смежных с ним бит на приемнике.

В качестве входных априорных данных алгоритм использует последовательность логарифмических метрик отношения правдоподобия (Log-Likelihood Ratios - LLR) $L^{pr}(x_i)$ для бит принятого декодируемого кодового слова:

$$L^{pr}(x_i) = \log \frac{\Pr(x_i = 0 | y_i)}{\Pr(x_i = 1 | y_i)}, \quad (1)$$

где x_i – i -ый бит исходного кодового слова на передатчике и y_i – его наблюдение на приемнике.

Каждая итерация BP-алгоритма состоит из двух шагов, последовательно обновляющих проверочные и кодовые узлы. Каждый проверочный узел CN_m генерирует «сообщение» $r_{m,n}(k)$ для каждого смежного с ним кодового узла VN_n следующим образом:

$$r_{m,n}(k) = \min_{j \in N_m \setminus n} (|q_{j,m}(k-1)|) \times \prod_{j \in N_m \setminus n} \text{sign}(q_{j,m}(k-1)), \quad (2)$$

где $q_{j,m}(k-1)$ – сообщение от кодового узла VN_j к проверочному узлу CN_m на $(k-1)$ -ой итерации. N_m обозначает подмножество кодовых узлов смежных с проверочным узлом CN_m . Следует отметить, что формула (2) является так называемой Min-Sum аппроксимацией оригинальной формулы BP-алгоритма [2]. Это позволяет значительно упростить аппаратную реализацию при незначительной деградации производительности алгоритма, и делает данный подход наиболее распространенным в практических реализациях декодера LDPC кодов.

Каждый кодовый узел VN_n обрабатывает сообщения $r_{m,n}(k)$, приходящие от смежных с ним проверочных узлов CN_m , и генерирует для каждого из них сообщение $q_{m,n}(k)$ следующим образом:

$$q_{m,n}(k) = L^{pr}(x_n) + \sum_{j \in M_n \setminus m} r_{j,n}(k), \quad (3)$$

где M_n – это подмножество проверочных узлов смежных с кодовым узлом VN_n .

На каждой итерации алгоритма происходит оценка переданных бит кодового слова на основе входных мягких метрик и сообщений от проверочных узлов:

$$\Lambda_n(k) = L^{pr}(x_n) + \sum_{m \in M_n} r_{m,n}(k), \quad (4)$$

где $\Lambda_n(k)$ – это LLR n -го бита декодируемого кодового слова на k -ой итерации алгоритма. Жесткое решение для n -го бита принятого кодового слова может быть получено как:

$$\hat{c}_n = \begin{cases} 0, & \text{если } \Lambda_n < 0 \\ 1, & \text{если } \Lambda_n > 0 \end{cases}. \quad (5)$$

Жесткое решение используется для вычисления синдрома ошибки кодового слова \mathbf{s} , который может быть найден как произведение проверочной матрицы \mathbf{H} и вектора-столбца декодируемого кодового слова \mathbf{c} :

$$\mathbf{s} = \mathbf{H}\mathbf{c}. \quad (6)$$

Алгоритм завершает работу, если значение синдрома ошибки \mathbf{s} равно нулевому вектору или при выполнении алгоритмом максимального количества итераций K_{max} .

D. Модифицированный алгоритм декодирования Layered Belief Propagation (LBP)

Для декодирования LDPC кодов, проверочная матрица которых обладает свойством слоистости, возможно применение модифицированного алгоритма Layered Belief Propagation (LBP). В этом случае одна итерация BP-алгоритма разбивается на несколько подитераций, число которых равно числу слоев проверочной матрицы p_{max} . В ходе выполнения одной подитерации алгоритма обновляется информация только для кодовых и проверочных узлов, относящихся к данному слою матрицы [1]. Таким образом, мягкая метрика оценки переданного бита $\Lambda_n^{(p)}$ на p -ой подитерации k -ой итерации алгоритма вычисляется как:

$$\Lambda_n^{(p)}(k) = \Lambda_n^{(p-1)}(k) - r_{m,n}(k-1) + r_{m,n}(k). \quad (7)$$

Подсчет синдромов ошибок производится на каждой подитерации алгоритма на основе оценок мягких метрик $\Lambda_n^{(p)}$. Условия завершения декодирования аналогичны BP-алгоритму. Скорость сходимости данного алгоритма выше, так как обмен сообщениями между узлами происходит не только между итерациями, но и между подитерациями алгоритма. Это улучшает корректирующую способность кода и уменьшает требуемое среднее число итераций декодирования.

III. ОБЗОР АППАРАТНЫХ АРХИТЕКТУР ДЕКОДЕРА LDPC КОДА СТАНДАРТА IEEE 802.11AD

A. Полностью параллельная архитектура

Полностью параллельная архитектура декодера не основывается на специальной структуре или свойствах проверочной матрицы кода и применима абсолютно для всех классов LDPC кодов, непосредственно реализуя BP-алгоритм. В этом случае каждому проверочному и кодовому узлу матрицы \mathbf{H} соответствует отдельный аппаратный блок, выполняющий необходимые операции на каждой итерации алгоритма. Общая структура полностью параллельного декодера представлена на рис. 3.

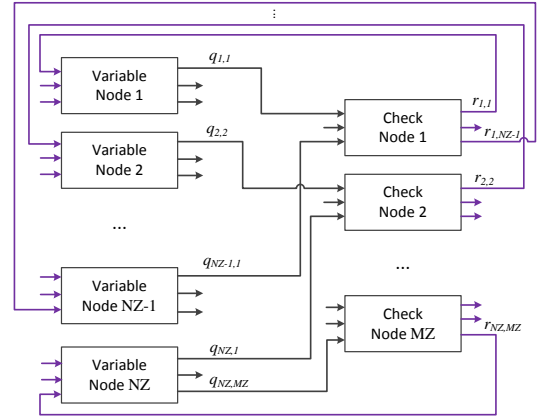


Рис. 3. Полностью параллельная архитектура LDPC декодера

Для рассматриваемого кода полностью параллельная архитектура использует 672 аппаратных кодовых блока (Variable Node) и 336 аппаратных проверочных блока (Check Node), выполняющих операции алгоритма декодирования, описанного в разделе 2.С. Для полностью параллельной архитектуры все сообщения проверочных и кодовых узлов вычисляются одновременно, что позволяет выполнять одну итерацию алгоритма за один такт работы цифрового аппаратного блока. Вследствие этого достигается наибольшая пропускная способность и наименьшее время задержки на декодирование. Недостатком данной архитектуры является то, что реализация каждого узла в виде отдельного аппаратного блока требует наибольшего количества аппаратных ресурсов и существенно усложняет процедуру трассировки декодера вследствие большого количества связей между блоками.

Следует отметить, что большинство практических задач не требуют очень высокой пропускной способности декодера, достигаемой полностью параллельной архитектурой, а существенные ограничения на объем используемых аппаратных ресурсов приводят к необходимости поиска альтернативных архитектур.

B. Слоистая архитектура декодера

Данная архитектура декодера реализует модифицированный алгоритм LBP, используя свойство слоистости проверочной матрицы \mathbf{H} , рассмотренное в разделе 2.В, которое улучшает скорость сходимости и позволяет переиспользовать аппаратные блоки между

подитерациями алгоритма. Согласно описанию из раздела 2.D на каждой подитерации алгоритма обновляются только те кодовые и проверочные узлы, которые относятся к текущему слою. Так как все слои рассматриваемого кода имеют вес столбца меньший либо равный 1, то каждая подитерация может быть гарантированно выполнена за один такт работы цифрового аппаратного блока, без возникновения коллизии при вычислении сообщений. Рассматриваемый код разбит на слои по строкам проверочной матрицы \mathbf{H} , что дает возможность переиспользовать между подитерациями аппаратные блоки проверочных узлов. Для рассматриваемого кода их требуемое число сокращено до 84, что в 4 раза меньше по сравнению со случаем полностью параллельной архитектуры. На рис. 4 представлена общая блок-схема слоистой архитектуры декодера.

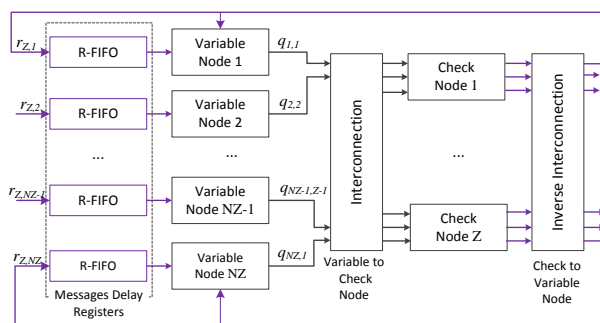


Рис. 4. Слоистая архитектура LDPC декодера

Данная архитектура декодера использует аппаратные блоки проверочных (Variable Node) и кодовых узлов (Check Node), прямой и обратный коммутаторы (Interconnection и Inverse Interconnection, соответственно) между двумя типами узлов, а также линии задержки сообщений (R-FIFO) от проверочных к кодовым узлам. Вследствие свойства слоистости каждый аппаратный кодовый узел ассоциирован только с одним аппаратным проверочным узлом. Это существенно упрощает аппаратную архитектуру кодового узла по сравнению с полностью параллельным случаем. Блок коммутации кодовых и проверочных узлов является переконфигурируемым, т.е. изменяющим связи между кодовыми и проверочными узлами каждый такт в зависимости от номера обрабатываемого слоя.

Анализ слоистой архитектуры показывает, что являясь концептуально более сложной по сравнению со случаем полностью параллельного декодера и требуя более сложной логики управления, данная архитектура все же позволяет добиться существенного сокращения общего числа используемых ресурсов ПЛИС за счет упрощения архитектуры кодовых узлов и уменьшения количества аппаратных проверочных узлов, что будет продемонстрировано далее.

С. Последовательно-параллельная архитектура декодера

Последовательно-параллельная архитектура является промежуточным вариантом между полностью параллельной и полностью последовательной архитектурами, доступной для рассматриваемого кода вследствие квазциклической структуры его проверочной матрицы.

На рис. 5 изображена архитектура последовательно-параллельного декодера, разработанного и анализируемого в данной работе. Архитектура основана на использовании LBP алгоритма декодирования. Для выполнения одной подитерации алгоритма, обновляющей сообщения проверочных и кодовых узлов, относящихся к текущему слою, требуется $2 \times N_{ne}(p)$ тактов, где $N_{ne}(p)$ – число элементов в p -ом слое макроматрицы больших либо равных нулю. $N_{ne}(p)$ тактов используется для обновления проверочных узлов и $N_{ne}(p)$ тактов используется для обновления кодовых узлов. Однако вследствие ортогональности двух строк макроматрицы рассматриваемого кода, формирующих один слой, становится возможным выполнение обновления кодовых и проверочных узлов параллельно, тем самым уменьшая эффективное время одной подитерации до $N_{ne}(p)$ тактов.

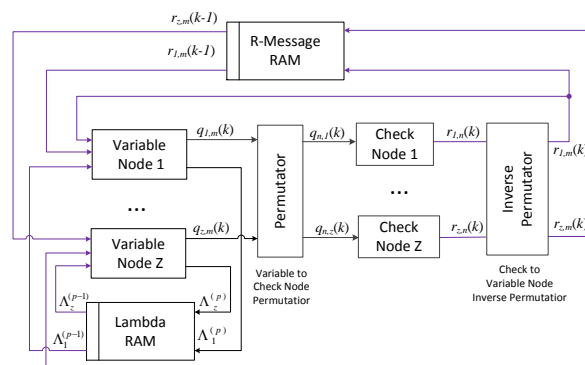


Рис. 5. Последовательно-параллельная архитектура LDPC декодера

Последовательно-параллельная архитектура использует Z аппаратных проверочных узлов (Check Node) и Z аппаратных кодовых узлов (Variable Node), соединенных между собой прямым и обратным коммутаторами (Permutator и Inverse Permutator соответственно). Связи блоков коммутации переконфигурируются в соответствие со значением циклического сдвига из макроматрицы QC-LDPC кода. Результаты вычислений сообщений $r_{m,n}(k)$ и мягких метрик кодового слова $\Lambda_n^{(p)}$ сохраняются в соответствующие блоки памяти (R-Message RAM и Lambda RAM, соответственно).

Размер элементарной матрицы Z QC-LDPC кода определяет степень параллелизма алгоритма, достигаемого последовательно-параллельной архитектурой декодера. Увеличение размера элементарной матрицы при фиксированной длине кодового слова позволяет уменьшить число тактов для вычисления одного слоя, тем самым увеличивая пропускную способность. Это, в свою очередь, ведет к росту числа используемых аппаратных ресурсов. Уменьшение величины Z дает обратный эффект уменьшения пропускной способности и используемой декодером аппаратной логики. Следует отметить, что данные рассуждения справедливы только в общем, так как изменение Z при сохранении длины кодового слова требует разработки новой макроматрицы и ведет к созданию нового кода, характеристики помехоустойчивости которого могут значительно отличаться. Для рассматриваемого кода Z равно 42,

что требует использования 42 аппаратных кодовых узлов и 42 аппаратных проверочных узлов, что, соответственно, в 16 и 8 раз меньше по сравнению с полностью параллельной архитектурой.

IV. СРАВНИТЕЛЬНЫЙ АНАЛИЗ АРХИТЕКТУР LDPC ДЕКОДЕРА

В данном разделе проведен сравнительный анализ трех рассмотренных ранее архитектур LDPC декодера с точки зрения достигаемой пропускной способности и требуемых аппаратных ресурсов. Все три рассмотренные варианта декодера были реализованы на языке описания аппаратуры Verilog, проверены функционально и синтезированы для технологии ПЛИС Xilinx Kintex 7.

В общем случае пропускная способность аппаратного LDPC декодера может быть оценена как [5]:

$$T = \frac{f \cdot R \cdot L}{N_{iter} \cdot S}, \quad (8)$$

где f – тактовая частота блока декодера, L – длина кодового слова в битах, R – скорость кодирования, N_{iter} – число итераций алгоритма декодирования, S – число тактов, требуемых для выполнения одной итерации.

Использование в качестве N_{iter} максимального числа итераций работы алгоритма K_{max} позволяет получить оценку нижнего порога пропускной способности. Однако возможность раннего завершения декодирования (при получении нулевого синдрома ошибки) приводит к тому, что в среднем для успешного декодирования кодового слова требуется меньшее число итераций, и влияние максимального числа итераций алгоритма как фактора, ограничивающего пропускную способность аппаратного декодера, можно существенно снизить. Известно, что требуемое среднее число итераций алгоритма зависит от рабочего отношения сигнал-шум (ОСШ). В данной работе для канала с аддитивным белым гауссовым шумом определялось ОСШ, соответствующее 1% отношения блочной ошибки декодирования (Block Error Rate - BLER). Для выбранного рабочего значения ОСШ определялось среднее число итераций, необходимое декодеру при условии раннего завершения декодирования. Полученное таким образом значение использовалось для расчета пропускной способности по формуле (8) и равнялось 5.1 для ВР-алгоритма и 4.1 для ЛВР-алгоритма.

Следует отметить, что формула (8) не учитывает время, необходимое на инициализацию декодера и завершение декодирования. Предполагается, что данное время оптимизировано для достижения пропускной способности, оцениваемой исходя из ограничений основного блока декодера.

Число используемых аппаратных ресурсов зависит не только от типа архитектуры декодера, но также от параметров целочисленной арифметики. Увеличение разрядности данных приводит к повышению точности целочисленной модели декодера, но и увеличивает аппаратные затраты на его реализацию. Для всех рассматриваемых реализаций декодера были выбраны единые параметры целочисленной арифметики, обеспечивающие деградацию рабочей точки на величину

не более чем 0.1 дБ ОСШ по сравнению с результатами моделирования декодера с использованием арифметики с плавающей точкой. Разрядность мягких метрик кодового слова $L^{pr}(k)$ и $\Lambda(k)$ равнялась 8 битам, разрядность сообщений между кодовыми и проверочными узлами $r_{m,n}(k)$ и $q_{n,m}(k)$ равнялась 5 битам.

Все рассматриваемые аппаратные архитектуры были реализованы на ПЛИС Xilinx Kintex7 XC7K325T. Частота тактирования декодера была выбрана одинаковой для всех рассматриваемых архитектур и равнялась 100 МГц. Оценка аппаратных ресурсов представлена как число используемых логических ячеек (LUT) и регистров ПЛИС. Общее число доступных аппаратных ресурсов для рассматриваемой ПЛИС составляет 203800 LUT и 407600 регистров, соответственно.

Результаты оценки пропускной способности и требуемых аппаратных ресурсов для всех рассмотренных архитектур декодера представлены в таблице 1. В данной работе в качестве дополнительного критерия аппаратной эффективности декодера предлагается отношение его пропускной способности к используемому числу логических ячеек, выраженное в единицах 1 Мбит/с на 1000 LUT. Данный критерий не учитывает число задействованных регистров ПЛИС, что представляется обоснованным, так как относительный процент использования регистров для всех рассматриваемых архитектур много меньше относительного числа используемых логических ячеек LUT.

Таблица 1

Результаты оценки пропускной способности и используемых аппаратных ресурсов

	Архитектура декодера		
	Полностью параллельная	Слоистая	Последовательно-параллельная
Число тактов для одной итерации алгоритма, S	1	4	52
Число итераций, N_{iter}	5.1	4.1	4.1
Пропускная способность декодера, Мбит/с	6 588	2 064	159
Число LUT, абсолютное	116 779	63 424	12 814
Число LUT, относительное, %	57.3	31.1	5.7
Число регистров, абсолютное	16 412	18 975	3 394
Число регистров, относительное %	4	4	0.8
Аппаратная эффективность декодера, Мбит/с / 1000LUT	56.4	32.5	12.4

Анализ полученных результатов показывает, что, предсказуемо, наибольшей пропускной способности равной 6.5 Гбит/с, а также наибольшего значения аппаратной эффективности, требующей наименьшего числа аппаратных ресурсов на 1 Мбит/с скорости декодирования, достигает полностью параллельная архитектура. Однако недостатком данного решения является невозможность ее масштабирования для меньших

значений пропускной способности. Исходя из этого, полностью параллельный декодер может быть рекомендован для тех задач, где необходимы максимальная производительность и минимальное время задержки на декодирование.

Декодер слоистой архитектуры требует 4 такта для выполнения одной итерации алгоритма, что уменьшает пропускную способность в 4 раза. Однако лучшая сходимость используемого LBP алгоритма приводит к уменьшению среднего числа итераций на 20%, что в итоге позволяет достигнуть пропускной способности 2.06 Гбит/с. Переиспользование аппаратных блоков проверочных узлов и упрощение архитектуры кодовых узлов уменьшает общее количество используемых LUT почти в два раза по сравнению с полностью параллельной архитектурой. Аппаратная эффективность декодера на 42% ниже по сравнению с предыдущим случаем, что позволяет рекомендовать слоистую архитектуру для применения в приложениях, где для достижения необходимой скорости декодирования (пропускной способности) применение данной архитектуры будет оказываться более эффективным, чем использование одного или нескольких декодеров с последовательно-параллельной архитектурой.

Последовательно-параллельная архитектура является наиболее компактной из всех рассматриваемых архитектур LDPC декодера. Данная реализация занимает 11% от общего объема LUT и достигает 2.4% пропускной способности относительно случая полностью параллельного декодера. Таким образом, аппаратная эффективность оказывается, соответственно, в 4.5 и 2.4 раза ниже относительно результатов полностью параллельной и слоистой архитектур. Требуемое число тактов для выполнения одной итерации алгоритма в 13 раз больше по сравнению со случаем слоистой архитектуры декодера, что соответствует пропускной способности равной 159 Мбит/с. Следует отметить, что в силу малого числа используемых аппаратных ресурсов пропускная способность может быть увеличена путем параллельного использования нескольких блоков декодера. Такой экстенсивный подход является оправданным до тех пор, пока количество требуемых аппаратных ресурсов не приблизится к результатам слоистой архитектуры декодера. Низкие требования по аппаратным ресурсам позволяют рекомендовать данную реализацию LDPC декодера для систем со средней и низкой пропускной способностью и строгими аппаратными ограничениями.

V. ЗАКЛЮЧЕНИЕ

LDPC коды являются одними из наиболее перспективных кодов исправляющих ошибки, благодаря их высокой помехоустойчивости и наличию подходов для эффективной аппаратной реализации алгоритмов декодирования. Существенным ограничением применимости данного класса кодов является отсутствие гибкости и масштабируемости базовых аппаратных архитектур декодера, что не позволяет разработчикам достигать необходимого компромисса между пропускной способностью и объемом используемых ресурсов. Эта проблема успешно решается для специально спроекти-

рованных кодов, свойства, структуры которых позволяют получать промежуточные аппаратные решения.

В данной работе исследованы реализации декодера LDPC кода из стандарта IEEE 802.11ad полностью параллельной, слоистой и последовательно-параллельной архитектур, выполнен анализ достигаемой пропускной способности и произведена оценка требуемого объема аппаратных ресурсов при реализации на ПЛИС. Эффективность реализации различных архитектур определялась на основе критерия отношения пропускной способности к объему используемых аппаратных ресурсов.

Результаты исследования показывают, что полностью параллельная архитектура, не требующая априорных сведений о структуре кода, позволяет достигать для рассматриваемого в данной работе кода пропускной способности декодера 6.5 Гбит/с и наибольшей аппаратной эффективности. Слоистая архитектура, основанная на возможности разложения проверочной матрицы \mathbf{H} на четыре слоя, строки которого ортогональны между собой, позволяет переиспользовать аппаратные блоки проверочных узлов, уменьшая на 46% число используемых ресурсов ПЛИС. При этом достигается пропускная способность 2.08 Гбит/с, что соответствует аппаратной эффективности на 42% ниже, по сравнению с полностью параллельной архитектурой. Использование последовательно-параллельной архитектуры позволяет реализовать декодер, использующий наименьшее количество аппаратных ресурсов, что составляет 11% от результата полностью параллельной архитектуры. Достигаемая пропускная способность при этом равна 159 Мбит/с.

ЛИТЕРАТУРА

- [1] IEEE P802.11ad: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications - Enhancements for Higher Throughput in the 60 GHz Band, IEEE, 2012.
- [2] Barry John R. Low-Density Parity-Check Codes. Georgia Institute of Technology. October 5, 2001.
- [3] Hocevar D.E. A reduced complexity decoder architecture via layered decoding of LDPC codes // Signal Processing Systems. 2004.
- [4] Kun Guo, Yong Hei and Shushan Qiao A Parallel-Layered Belief-Propagation Decoder for Non-layered LDPC Codes // Journal of Communications. May 2010. Vol 5. № 5. P. 400-408.
- [5] Wiener M., Nikolic B. A high-throughput, flexible LDPC decoder for multi-Gbps Wireless Personal Area Networks. UCB technical report. December 2010.
- [6] Zarubica R., Wilson S.G., Hall E. Multi-Gbps FPGA-Based Low Density Parity Check (LDPC) Decoder Design // Proceedings of the IEEE Global Telecommunications Conference. Washington, DC. 26-30 November 2007. P. 548-552.
- [7] Blanksby A., Shen B., Trachewsky J. LDPC code set for mmWave communication // Proceedings of the 2010 ACM international workshop on mmWave communications. Sep. 2010. P. 39-43.